

Application of the new IAPWS Guideline on the fast and accurate calculation of steam and water properties with the Spline-Based Table Look-Up Method (SBTL) in RELAP-7

R. A. Berry, H.-J. Kretzschmar, U.
Gampe, R. C. Martineau, M. Kunick

July 2017

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance



**Application of the new IAPWS Guideline on the fast
and accurate calculation of steam and water
properties with the Spline-Based Table Look-Up
Method (SBTL) in RELAP-7**

R. A. Berry, H.-J. Kretzschmar, U. Gampe, R. C. Martineau, M. Kunick

July 2017

**Idaho National Laboratory
Idaho Falls, Idaho 83415**

<http://www.inl.gov>

**Prepared for the
U.S. Department of Energy**

**Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517**

Application of the new IAPWS Guideline on the fast and accurate calculation of steam and water properties with the Spline-Based Table Look-Up Method (SBTL) in RELAP-7

The numerical simulation of thermohydraulic processes in nuclear power plants requires very accurate and extremely fast algorithms for calculating the thermophysical properties of water and steam. In order to provide such algorithms, the International Association for the Properties of Water and Steam (IAPWS) has adopted the new “IAPWS Guideline on the Fast Calculation of Steam and Water Properties with the Spline-Based Table Look-Up Method (SBTL)” [1]. In this article, the SBTL method is applied to property functions of specific volume and specific internal energy (v, e) based on the scientific formulation IAPWS-95 and the latest IAPWS formulations for transport properties. From the newly generated SBTL functions, thermodynamic and transport properties as well as their derivatives and inverse functions are calculable in the fluid range of state for pressures up to 100 MPa and for temperatures up to 1273 K, including the metastable liquid and the metastable vapor regions. The SBTL functions reproduce the underlying formulations with an accuracy of 10-100 ppm and significantly reduced computing times. The SBTL method has been implemented into the nuclear reactor system safety analysis code RELAP-7 [2] to consider the real fluid behavior of water and steam in a novel 7-equation two-phase flow model.

Anwendung der neuen IAPWS-Richtlinie zur schnellen und genauen Berechnung der Eigenschaften von Wasser und Wasserdampf mit dem Spline-Basierten Table Look-up Verfahren (SBTL) in RELAP-7

Die numerische Simulation thermohydraulischer Prozesse in Kernkraftwerksanlagen erfordert sehr genaue und extrem schnelle Stoffwert-Berechnungsalgorithmen für Wasser und Wasserdampf. Zu diesem Zweck hat die International Association for the Properties of Water and Steam (IAPWS) die neue “IAPWS Guideline on the Fast Calculation of Steam and Water Properties with the Spline-Based Table Look-Up Method (SBTL)” [1] verabschiedet. In diesem Beitrag wird das SBTL Verfahren auf Stoffwertfunktionen von spezifischem Volumen und spezifischer innerer Energie (v, e) basierend auf der wissenschaftlichen Formulierung IAPWS-95 und den neuesten IAPWS Standards für Transporteigenschaften angewendet. Mit den neuen SBTL Funktionen lassen sich die thermodynamischen Zustandsgrößen und Transporteigenschaften sowie deren Ableitungen und Umkehrfunktionen im fluiden Zustandsgebiet bei Drücken bis zu 100 MPa und Temperaturen bis zu 1273 K, inklusive der metastabilen Gebiete für überhitzte Flüssigkeit und unterkühltes Gas, berechnen. Die SBTL Funktionen geben die zugrundeliegenden Formulierungen mit einer Genauigkeit von 10-100 ppm und erheblich reduzierten Rechenzeiten wieder. Das SBTL Verfahren wurde in den zur Sicherheitsanalyse von Kernreaktorsystemen entwickelten Code RELAP-7 [2] implementiert um das reale Zustandsverhalten von Wasser und Wasserdampf in einem neuen 7-Gleichungsmodell für die Zweiphasenströmung zu berücksichtigen.

1 Introduction

The analysis of thermohydraulic processes in nuclear power plants with numerical simulations is of particular importance for safety assessment and the development of advanced technologies. For these simulations, system codes and Computational Fluid Dynamics (CFD) are widely in use. In particular, the detailed analysis of transient processes is computationally intensive and requires long computing times. A large portion of the computing time is demanded for the calculation of fluid properties. In the thermohydraulic system code RELAP-7 [2] as well as in density based CFD solvers, fluid properties are most frequently calculated from specific volume and specific internal energy (v,e). Furthermore, numerically consistent inverse functions of pressure and temperature (p,T), pressure and specific volume (p,v), pressure and specific enthalpy (p,h), pressure and specific entropy (p,s), and specific enthalpy and specific entropy (h,s) are required. The calculation of these property functions from accurate fundamental equations of state, such as the IAPWS Formulation 1995 for General and Scientific Use (IAPWS-95) [3, 4] for water and steam, requires iterative algorithms leading to unacceptable computing times. For faster property calculations, the IAPWS Industrial Formulation 1997 (IAPWS-IF97) [5, 6] and its supplementary releases on backward equations [7, 8, 9, 10] enable computations from (p,T), (p,h), (p,s), and (h,s) without iterative procedures. Due to the imperfect numerical consistency with the basic equations of IAPWS-IF97, the application of backward equations for simulating processes with small spatial and time discretization can lead to convergence problems. In these situations, inverse functions must be calculated by iteration from the basic equations with starting values determined from the available backward equations. Backward equations do not exist for functions of (v,e) and (p,v) or for fluids other than water and steam. Therefore, property calculations are often simplified, for instance with cubic equations of state or with the ideal-gas model. This leads, depending on the range of state, to inaccurate simulation results. To calculate fluid properties more accurately but with reasonable computing times, so-called table look-up methods are frequently applied. For these methods, the desired fluid properties are interpolated from previously tabulated values, which are calculated from an accurate property formulation. The Spline-Based Table Look-up method (SBTL) has been developed in a project of the International Association for the Properties of Water and Steam (IAPWS) and was adopted as a Guideline [1, 11, 12] in 2015. This method has been applied successfully in the CFD code TRACE of the German Aerospace Agency (DLR) [13] and other codes for process simulations. In this work, the SBTL method is applied to IAPWS-95 to consider the real fluid behavior of water and steam in RELAP-7 [2]. From the newly generated SBTL functions, thermodynamic and transport properties as well as their derivatives are calculable. The range of validity covers the stable fluid phases for pressures up to 100 MPa and temperatures up to 1273 K as well as metastable states in the superheated liquid phase and the subcooled vapor phase. The latter is of particular importance to enable the 7-equation two-phase flow model in RELAP-7 [2] and to consider the delayed condensation in the simulation of low-pressure steam turbines in CFD. Furthermore, numerically consistent inverse functions for various input variable combinations are provided.

2 List of symbols

Symbol	Quantity
a	Spline polynomial coefficient
c_p	Specific isobaric heat capacity
c_v	Specific isochoric heat capacity
f	Function
floor()	Round down
\mathbf{F}	Vector of functions
g	Specific Gibbs free energy
h	Specific enthalpy
i	Interval index; Cell index in v direction
I	Number of nodes along \bar{p} for $T_s(\bar{p})$ or along \bar{v} for $p(\bar{v}, e)$
j	Cell index in e direction
J	Number of nodes along e
\mathbf{J}	Jacobian matrix
k	Serial number; Exponent
l	Serial number; Exponent
p	Pressure
\bar{p}	Transformed pressure
s	Specific entropy
T	Temperature
e	Specific internal energy
v	Specific volume
\bar{v}	Transformed specific volume
w	Speed of sound
x	Vapor fraction
\mathbf{X}	Vector of unknowns
η	Dynamic viscosity
λ	Thermal conductivity

Superscript	
AUX	Auxiliary spline function
G	Spline function for the gas region
INV	Inverse spline function
K	Knot
L	Spline function for the liquid region
T	Transposed
'	Saturated-liquid state
“	Saturated-vapor state

Subscript	
i	Cell index in \bar{v} direction
j	Cell index in e direction
liq_spin	At the liquid spinodal
s	At saturation

3 The Spline-Based Table Look-up method (SBTL) and its application to IAPWS-95

For fast and accurate property calculations in numerical process simulations, table look-up methods are frequently applied. For these methods, discrete values of the required properties are calculated from accurate property formulations and are stored as nodes in look-up tables. During the process simulation, properties are determined from the tabulated values with simple interpolation or approximation algorithms. With local algorithms, such as bi-linear or local bi-cubic interpolation, properties within a cell of the look-up table are calculated from adjacent values only, as opposed to global methods, where all tabulated values are used. Although local algorithms are easily implemented, they do have their deficiencies. Bi-linear interpolation requires comparatively large look-up tables to represent the fluid properties with acceptable accuracy and does not provide continuous first derivatives. Local bi-cubic interpolation overcomes these problems, but the calculation of inverse functions is computationally intensive, as it requires the evaluation of trigonometric functions. Moreover, to achieve the desired accuracy over the entire range of state, the look-up tables are frequently prepared with variable distances between the nodes. This requires extensive cell search algorithms during the property calculation and decreases the computing speed.

In order to provide fast and accurate property calculation algorithms for computationally intensive process simulations, the Spline-Based Table Look-up Method (SBTL) [1, 11, 12] was developed in a project of the International Association for the Properties of Water and Steam (IAPWS). Spline functions are continuous, piecewise defined functions. For pure fluids or mixtures at constant composition, phase boundaries, such as $T_s(p)$, are represented by one-dimensional spline functions, whereas the properties in the single-phase region, such as $p(v,e)$, can be described by two-dimensional spline functions. For mixtures with variable composition, the method can be extended using multivariate spline interpolation. Alternatively, a mixture model can be used, where each pure component is described separately. In most numerical process simulations, property functions need to be continuously differentiable once. Global spline interpolation algorithms with second-degree polynomials are capable to fulfil this requirement. By means of variable transformations, the function to be interpolated is linearized. The spline function is constructed on piecewise equidistant nodes in transformed coordinates. This results in simple search algorithms to find the interval or cell in the look-up table, which corresponds to the given input variables. In this way, the underlying property formulations can be reproduced with high accuracy and low computing time. Moreover, second-degree polynomials can easily be solved analytically in terms of the independent variables. This enables the fast calculation of numerically consistent inverse functions.

In this work, the SBTL method is applied to property functions for water and steam based on IAPWS-95 for the range of validity outlined in Section 3.1. In Sections 3.2 - 3.5, the basic principles of the SBTL method are described for one- and two-dimensional spline functions. In Sections 3.6 - 3.9, the newly generated SBTL functions are presented.

3.1 Range of validity

The range of validity of the generated SBTL property functions is subdivided into the liquid region L, the gas region G, and the stable two-phase region TP. It is bounded as follows:

$$273.15 \text{ K} \leq T \leq 1273.15 \text{ K} \quad 611.212677 \text{ Pa} \leq p \leq 100 \text{ MPa}.$$

The lower pressure limit is set to $p_s(273.15 \text{ K}) = 611.212677 \text{ Pa}$. Figure 1 shows the liquid region L, which also includes the metastable superheated liquid phase between the saturated liquid curve ($x=0$) and the liquid spinodal. Analogously, the gas region G includes the metastable subcooled vapor phase between the saturated vapor curve ($x=1$) and the vapor spinodal as shown in Figure 2. The boundary between the liquid region L and the gas region G is defined by the specific internal energy at the critical point $e_c = 2015.734524 \text{ kJ/kg}$.

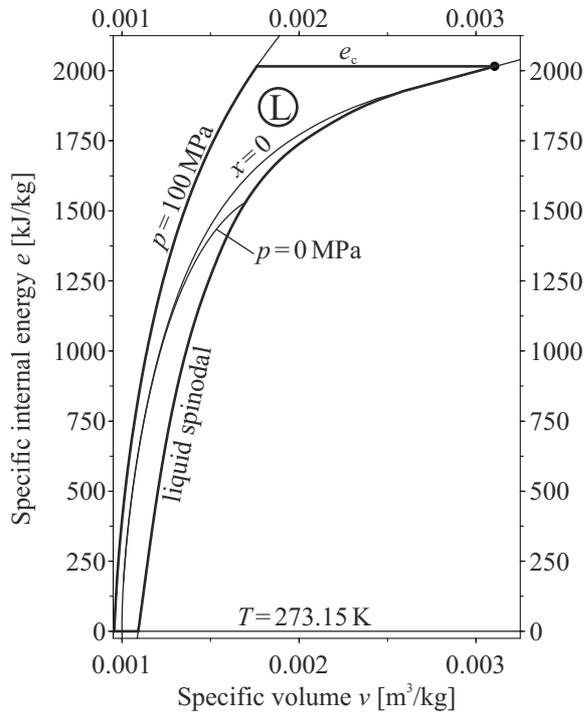


Figure 1: Region L in the v - e plane.

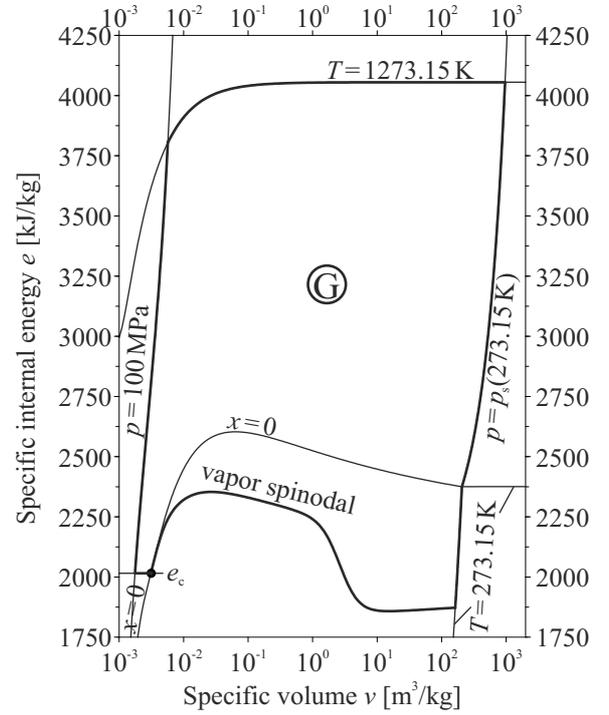


Figure 2: Region G in the v - e plane.

3.2 One-dimensional spline functions using the example of $T_s(p)$

In this example for one-dimensional spline functions, a quadratic spline function for $T_s(p)$ is generated. In order to enhance the accuracy of any interpolation or approximation method, it is advantageous to linearize the function to be interpolated first by means of suitable variable transformations. For many saturation-pressure equations, temperature and pressure are transformed as $\bar{T} = T^{-1}$ and $\bar{p} = \ln(p)$. The computation of the natural logarithm is computationally intensive since it is not an algebraic function. For a computationally more efficient linearization, the pressure is transformed using $\bar{p} = \sqrt[4]{p}$ and the temperature transformation is omitted. In this way, the change in curvature, which cannot be reproduced by a quadratic polynomial, is reduced and the resulting spline function is more accurate. The effect of this simple variable transformation is shown in Figs. 3 and 4.

The spline function is created in transformed coordinates and interpolates values between a series of I discrete data points, the so-called nodes (see Fig. 4). The $T_{s,i}(\bar{p}_i)$ values at the nodes are calculated from the underlying equation of state. The I spline polynomials are connected at knots, which are located in the middle between the nodes along \bar{p} . The \bar{p}_i^K values of the $I+1$ knots are

$$\bar{p}_{i+1}^K = \frac{1}{2}(\bar{p}_i + \bar{p}_{i+1}), \quad i = 1, \dots, I-1 \quad (1.1)$$

$$\bar{p}_1^K = \bar{p}_1 - \frac{1}{2}(\bar{p}_2 - \bar{p}_1), \text{ and} \quad \bar{p}_{I+1}^K = \bar{p}_I + \frac{1}{2}(\bar{p}_I - \bar{p}_{I-1}). \quad (1.2, 1.3)$$

The T_s positions of the knots result from the spline algorithm as explained below. A spline polynomial ranges over the interval $\{i\}$ between two knots

$$\bar{p}_i^K \leq \bar{p} < \bar{p}_{i+1}^K \quad (1.4)$$

and intersects the node (i) at $T_{s,i}(\bar{p}_i)$. The number of I nodes is chosen to ensure the required accuracy of the spline function over its full domain of definition. The nodes are distributed equidistantly along \bar{p} so that a simple search algorithm can be used to determine the interval $\{i\}$ in the series of knots that fulfills Eq. 1.4 for a given transformed variable \bar{p} . For equidistant nodes, and therefore equidistant knots, i can easily be calculated from

$$i = \text{floor} \left(\frac{\bar{p} - \bar{p}_1^K}{\Delta \bar{p}^K} \right). \quad (1.5)$$

The distribution of nodes and knots can also be manipulated by piecewise equidistant nodes, in ranges for which $\Delta \bar{p} = \bar{p}_{i+1} - \bar{p}_i$ is constant. In each interval $\{i\}$ a quadratic polynomial

$$T_{s,\{i\}}(\bar{p}) = \sum_{k=1}^3 a_{ik} (\bar{p} - \bar{p}_i)^{k-1}, \quad (1.6)$$

is defined. The $3I$ coefficients a_{ik} of the I spline polynomials are obtained from the following conditions. Each of the I polynomials $T_{s,\{i\}}(\bar{p})$ intersects the node (i)

$$T_{s,\{i\}}(\bar{p}_i) = T_{s,i}(\bar{p}_i), \quad i = 1, \dots, I. \quad (1.7)$$

In addition, the T_s values at the inner $I-1$ knots are equal for the adjacent polynomials

$$T_{s,\{i\}}(\bar{p}_{i+1}^K) = T_{s,\{i+1\}}(\bar{p}_{i+1}^K), \quad i = 1, \dots, I-1. \quad (1.8)$$

The first derivatives with respect to \bar{p} at each of these knots are also equal

$$\left. \frac{dT_s}{d\bar{p}} \right|_{\{i\}} (\bar{p}_{i+1}^K) = \left. \frac{dT_s}{d\bar{p}} \right|_{\{i+1\}} (\bar{p}_{i+1}^K), \quad i = 1, \dots, I-1. \quad (1.9)$$

At the outer knots, these derivatives are calculated from the underlying function $T_s(p)$ with

$$\left. \frac{dT_s}{d\bar{p}} \right|_{\{i=1\}} (\bar{p}_1^K) = \frac{dT_s}{d\bar{p}} (\bar{p}_1^K) \text{ and} \quad \left. \frac{dT_s}{d\bar{p}} \right|_{\{i=I\}} (\bar{p}_{I+1}^K) = \frac{dT_s}{d\bar{p}} (\bar{p}_{I+1}^K), \quad (1.10, 1.11)$$

where

$$\frac{dT_s}{d\bar{p}} = \frac{dT_s}{dp} \frac{dp}{d\bar{p}}.$$

The linear system of Eqs. (1.7 - 1.11) is solved in order to obtain the $3I$ spline coefficients a_{ik} . The resulting spline function and its first derivative is continuous over its full domain of definition. A more general description of the SBTL method is given in [1, 11, 12].

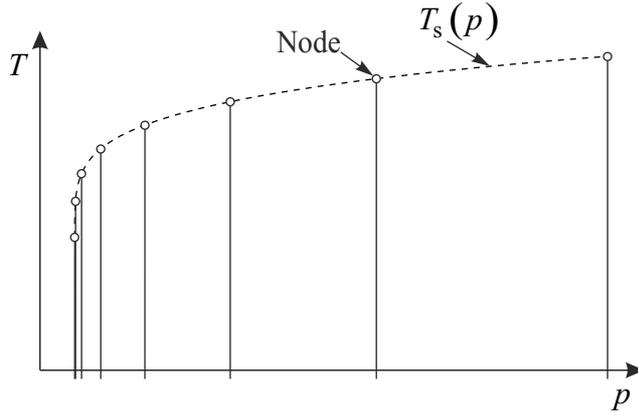


Figure 3: Untransformed function $T_s(p)$ with nodes equidistant in \bar{p} , rather than in p .

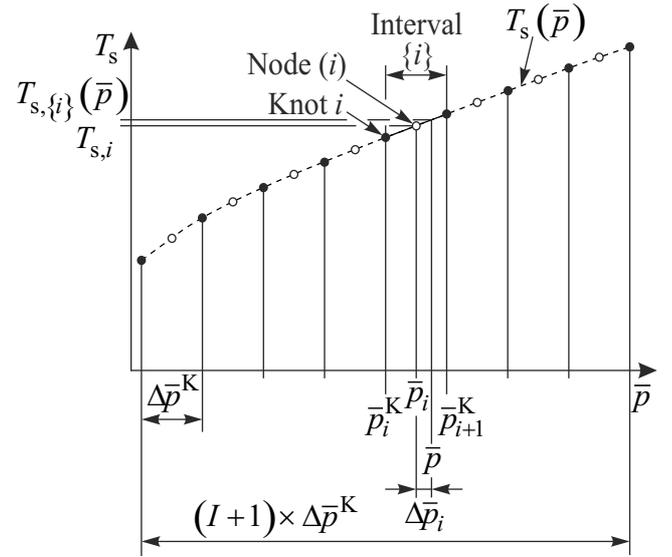


Figure 4: Transformed function $T_s(\bar{p})$ with nodes equidistant in \bar{p} .

3.3 Calculation of inverse functions using the example of $p_s(T)$

The simulation of processes with small spatial and time discretization requires the inverse functions to be numerically consistent with their corresponding forward functions. The utilization of quadratic spline polynomials enables the fast calculation of numerically consistent inverse functions. For example, from the spline function for $T_s(p)$ an inverse spline function for $p_s(T)$ can be obtained by solving Eq. (1.6) in terms of the transformed pressure \bar{p} with

$$\bar{p}_{s,\{i\}}^{\text{INV}}(T) = \frac{(-B_i \pm \sqrt{B_i^2 - 4A_i C_i(T)})}{2A_i} + \bar{p}_{s,i}, \quad (1.12)$$

where

$$A_i = a_{i3},$$

$$B_i = a_{i2}, \text{ and}$$

$$C_i(T) = a_{i1} - T.$$

For the monotonic function $T_{s,\{i\}}(\bar{p})$ in the interval $\{i\}$, the sign (\pm) in Eq. (1.12) equals $\text{sgn}(B_i) = +$. In order to calculate $p_s(T)$ from Eq. (1.12), the corresponding interval $\{i\}$, where the given temperature T is located, must be determined. For this purpose, the auxiliary spline function $\bar{p}_s^{\text{AUX}}(T)$ and Eq. (1.5) are used. Then, the inverse spline polynomial $\bar{p}_{s,\{i\}}^{\text{INV}}(T)$, Eq. (1.12), is evaluated. The result must fulfil the condition described by Eq. (1.4); otherwise, the index i needs to be incremented or decremented and the calculation repeated. Eventually, \bar{p}_s is converted to p_s with the inverse transformation function $p_s(\bar{p}_s)$. A comprehensive description of the calculation of inverse functions from monotonic and non-monotonic spline functions is given in [1, 11, 12].

3.4 Two-dimensional spline functions using the example of $p(v,e)$

Analogously to the one-dimensional case discussed in Section 3.2, the generation of a two-dimensional bi-quadratic spline function is illustrated using the example of $p(v,e)$. Two separate spline functions for the liquid region L and the gas region G are prepared. Again, variable transformations are applied to reduce the change in curvature, i.e., to minimize the third derivatives that cannot be reproduced by bi-quadratic polynomials. For this purpose, both, the dependent and the independent variables can be transformed. Thus, the bi-quadratic spline polynomials can reproduce the transformed property function more accurately. In the gas phase, the changes in p are almost proportional to changes in e . Therefore, a transformation of e is not required. Depending on the desired range of validity, suitable variable transformations of v in the gas phase can be either $\bar{v} = v^{-1}$, $\bar{v} = \sqrt[4]{v}$, or $\bar{v} = \ln(v)$. A transformation of the dependent variable p is also possible, but is not applied in this example. For some functions, a combination with the independent variables, e.g. $\bar{p} = pv$, is useful to efficiently linearize the property function. Variable transformations can also be utilized to distribute the nodes in the desired range of validity more efficiently. Through the use of a so-called scaling transformation, the irregularly shaped range of validity of the liquid region L, see Fig. 5, is mapped onto a rectangle, see Fig. 6. The linear scaling function reads

$$\bar{v}(v,e) = \frac{\bar{v}_{\max} - \bar{v}_{\min}}{v_{\max}(e) - v_{\min}(e)} \cdot (v - v_{\min}(e)) + \bar{v}_{\min}, \quad (2.1)$$

where $v_{\min}(e) = v(p_{\max}, e)$ and $v_{\max}(e) = v_{\text{liq_spin}}(e)$ are represented by quadratic spline functions and the free parameters are set to $\bar{v}_{\min} = 1$ and $\bar{v}_{\max} = 100$.

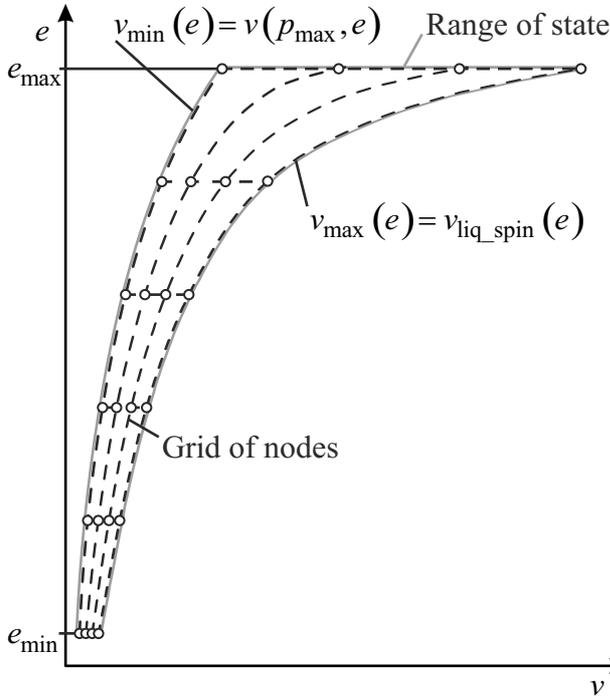


Figure 5: Grid of nodes in untransformed coordinates.

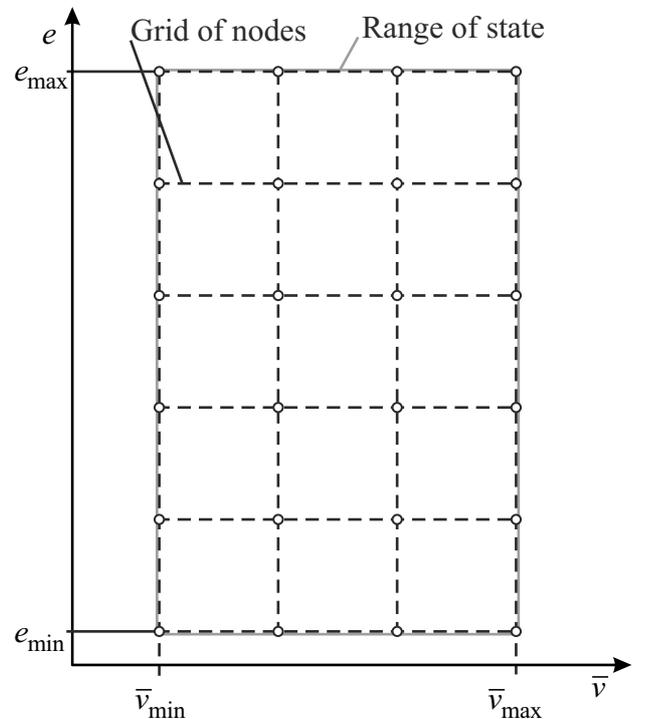


Figure 6: Grid of nodes in transformed coordinates.

Each spline function, $p^L(v,e)$ and $p^G(v,e)$, is created in transformed coordinates and interpolates values between a grid of IJ discrete nodes (see Fig. 7). The $p_{i,j}(\bar{v}_i, e_j)$ values at the nodes are calculated from the underlying equation of state. The IJ spline polynomials are connected at knots, which are located in the middle between the nodes along \bar{v} and e . The \bar{v}_i^K and e_j^K values of the $(I+1)(J+1)$ knots are

$$\bar{v}_{i+1}^K = \frac{1}{2}(\bar{v}_i + \bar{v}_{i+1}), \quad i = 1, \dots, I-1 \quad (2.2)$$

$$\bar{v}_1^K = \bar{v}_1 - \frac{1}{2}(\bar{v}_2 - \bar{v}_1), \quad \bar{v}_{I+1}^K = \bar{v}_I + \frac{1}{2}(\bar{v}_I - \bar{v}_{I-1}), \quad (2.3, 2.4)$$

$$e_{j+1}^K = \frac{1}{2}(e_j + e_{j+1}), \quad i = 1, \dots, J-1 \quad (2.5)$$

$$e_1^K = e_1 - \frac{1}{2}(e_2 - e_1), \quad \text{and} \quad e_{J+1}^K = e_J + \frac{1}{2}(e_J - e_{J-1}). \quad (2.6, 2.7)$$

The p positions of the knots result from the spline algorithm as explained below. A spline polynomial ranges over the cell $\{i,j\}$ between

$$\bar{v}_i^K \leq \bar{v} < \bar{v}_{i+1}^K \quad \text{and} \quad e_j^K \leq e < e_{j+1}^K \quad (2.8, 2.9)$$

and intersects the node (i,j) at $p_{i,j}(\bar{v}_i, e_j)$.

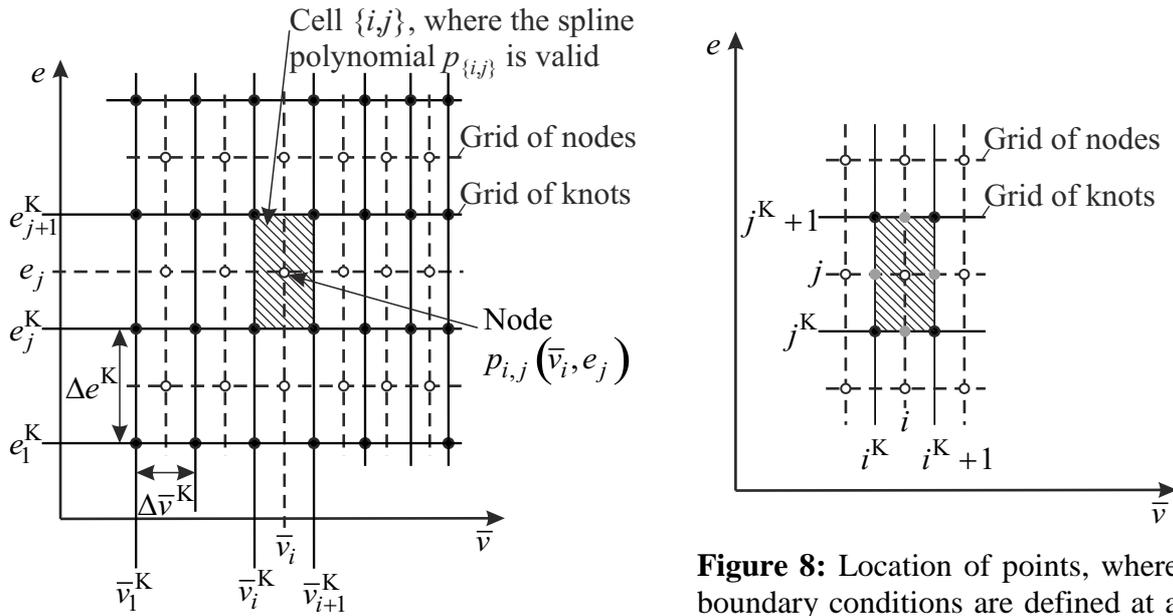


Figure 7: Grid of nodes and grid of knots with cell $\{i,j\}$, where the spline polynomial $p_{\{i,j\}}(\bar{v}, e)$ is valid.

Figure 8: Location of points, where boundary conditions are defined at a cell.

The number of IJ nodes is chosen to ensure the required accuracy of the spline function over its full domain of definition. The nodes are distributed equidistantly along \bar{v} and e , so that a simple search algorithm can be used to determine the cell $\{i,j\}$ in the grid of knots that fulfills the Eqs. (2.8) and (2.9) for a given pair of transformed variables (\bar{v}, e) . For equidistant nodes, and therefore equidistant knots, i and j can easily be calculated from

$$i = \text{floor} \left(\frac{\bar{v} - \bar{v}_1^K}{\Delta \bar{v}^K} \right) \text{ and} \quad (2.10)$$

$$j = \text{floor} \left(\frac{e - e_1^K}{\Delta e^K} \right). \quad (2.11)$$

The distribution of nodes and knots can also be manipulated by piecewise equidistant nodes, in ranges for which $\Delta \bar{v} = \bar{v}_{i+1} - \bar{v}_i$, or $\Delta e = e_{j+1} - e_j$ respectively, are constant. In each cell $\{i, j\}$ a bi-quadratic polynomial

$$p_{\{i, j\}}(\bar{v}, e) = \sum_{k=1}^3 \sum_{l=1}^3 a_{ijkl} (\bar{v} - \bar{v}_i)^{k-1} (e - e_j)^{l-1} \quad (2.12)$$

is defined. The $9IJ$ coefficients a_{ijkl} of the IJ spline polynomials are obtained from the conditions described in the following. Figure 8 illustrates the location of points, where boundary conditions are defined at a cell. Each of the IJ polynomials $p_{\{i, j\}}(\bar{v}, e)$ intersects the node (i, j)

$$p_{\{i, j\}}(\bar{v}_i, e_j) = p_{i, j}(\bar{v}_i, e_j) \quad i = 1, \dots, I, \quad \left(i^K, j \right), \quad \left(i^K + 1, j \right), \quad \left(i, j^K \right) \quad (2.13)$$

The p values at the midpoints of the cell boundaries $\left(i, j + \frac{1}{2} \right)$, marked with gray circles in Fig. 8, are equal to the corresponding values of the adjacent cells

$$p_{\{i, j\}}(\bar{v}_{i+1}^K, e_j) = p_{\{i+1, j\}}(\bar{v}_{i+1}^K, e_j) \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (2.14)$$

$$p_{\{i, j\}}(\bar{v}_i, e_{j+1}^K) = p_{\{i, j+1\}}(\bar{v}_i, e_{j+1}^K) \quad i = 1, \dots, I, \quad j = 1, \dots, J-1. \quad (2.15)$$

Furthermore, the derivatives $\left(\frac{\partial p}{\partial \bar{v}} \right)_e$ at $\left(i^K, j \right)$ and $\left(i^K + 1, j \right)$, as well as $\left(\frac{\partial p}{\partial e} \right)_{\bar{v}}$ at $\left(i, j + \frac{1}{2} \right)$ and $\left(i, j + 1 \right)$, are equal to the corresponding derivatives of the adjacent cells

$$\left(\frac{\partial p}{\partial \bar{v}} \right)_e \Big|_{\{i, j\}}(\bar{v}_{i+1}^K, e_j) = \left(\frac{\partial p}{\partial \bar{v}} \right)_e \Big|_{\{i+1, j\}}(\bar{v}_{i+1}^K, e_j) \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (2.16)$$

$$\left(\frac{\partial p}{\partial e} \right)_{\bar{v}} \Big|_{\{i, j\}}(\bar{v}_i, e_{j+1}^K) = \left(\frac{\partial p}{\partial e} \right)_{\bar{v}} \Big|_{\{i, j+1\}}(\bar{v}_i, e_{j+1}^K) \quad i = 1, \dots, I, \quad j = 1, \dots, J-1. \quad (2.17)$$

In addition, the p values and the crossed derivatives $\left(\frac{\partial^2 p}{\partial \bar{v} \partial e} \right)$ at the four knots at the corners $\left(i^K, j^K \right)$, $\left(i^K, j^K + 1 \right)$, $\left(i^K + 1, j^K \right)$, and $\left(i^K + 1, j^K + 1 \right)$ are equal to the corresponding values of the neighboring cells

$$p_{\{i, j\}}(\bar{v}_{i+1}^K, e_j^K) = p_{\{i+1, j\}}(\bar{v}_{i+1}^K, e_j^K) \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (2.18)$$

$$p_{\{i, J\}}(\bar{v}_{i+1}^K, e_{J+1}^K) = p_{\{i+1, J\}}(\bar{v}_{i+1}^K, e_{J+1}^K) \quad i = 1, \dots, I-1, \quad (2.19)$$

$$p_{\{i, j\}}(\bar{v}_i, e_{j+1}^K) = p_{\{i, j+1\}}(\bar{v}_i, e_{j+1}^K) \quad i = 1, \dots, I, \quad j = 1, \dots, J-1, \quad (2.20)$$

$$p_{\{I, j\}}(\bar{v}_{I+1}^K, e_{j+1}^K) = p_{\{I, j+1\}}(\bar{v}_{I+1}^K, e_{j+1}^K) \quad j = 1, \dots, J-1, \quad (2.21)$$

$$\frac{\partial^2 p}{\partial \bar{v} \partial e} \Big|_{\{i, j\}}(\bar{v}_{i+1}^K, e_j^K) = \frac{\partial^2 p}{\partial \bar{v} \partial e} \Big|_{\{i+1, j\}}(\bar{v}_{i+1}^K, e_j^K) \quad i = 1, \dots, I-1, \quad j = 1, \dots, J, \quad (2.22)$$

$$\frac{\partial^2 p}{\partial \bar{v} \partial e} \Big|_{\{i, J\}}(\bar{v}_{i+1}^K, e_{J+1}^K) = \frac{\partial^2 p}{\partial \bar{v} \partial e} \Big|_{\{i+1, J\}}(\bar{v}_{i+1}^K, e_{J+1}^K) \quad i = 1, \dots, I-1, \quad (2.23)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{i,j\}} (\bar{v}_i^K, e_{j+1}^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{i,j+1\}} (\bar{v}_i^K, e_{j+1}^K) \quad i=1, \dots, I, \quad j=1, \dots, J-1, \quad (2.24)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,j\}} (\bar{v}_{I+1}^K, e_{j+1}^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,j+1\}} (\bar{v}_{I+1}^K, e_{j+1}^K) \quad j=1, \dots, J-1, \quad (2.25)$$

At the outer boundaries of the grid of knots, the following values are provided from the underlying equation of state

$$\left(\frac{\partial p}{\partial \bar{v}} \right)_e \Big|_{\{1,j\}} (\bar{v}_1^K, e_j) = \left(\frac{\partial p}{\partial \bar{v}} \right)_e (\bar{v}_1^K, e_j) \quad j=1, \dots, J, \quad (2.26)$$

$$\left(\frac{\partial p}{\partial \bar{v}} \right)_e \Big|_{\{I,j\}} (\bar{v}_{I+1}^K, e_j) = \left(\frac{\partial p}{\partial \bar{v}} \right)_e (\bar{v}_{I+1}^K, e_j) \quad j=1, \dots, J, \quad (2.27)$$

$$\left(\frac{\partial p}{\partial e} \right)_{\bar{v}} \Big|_{\{i,1\}} (\bar{v}_i, e_1^K) = \left(\frac{\partial p}{\partial e} \right)_{\bar{v}} (\bar{v}_i, e_1^K) \quad i=1, \dots, I, \quad (2.28)$$

$$\left(\frac{\partial p}{\partial e} \right)_{\bar{v}} \Big|_{\{i,J\}} (\bar{v}_i, e_{J+1}^K) = \left(\frac{\partial p}{\partial e} \right)_{\bar{v}} (\bar{v}_i, e_{J+1}^K) \quad i=1, \dots, I, \quad (2.29)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{1,1\}} (\bar{v}_1^K, e_1^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{1,1\}} (\bar{v}_1^K, e_1^K), \quad (2.30)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,1\}} (\bar{v}_{I+1}^K, e_1^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,1\}} (\bar{v}_{I+1}^K, e_1^K), \quad (2.31)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{1,J\}} (\bar{v}_1^K, e_{J+1}^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{1,J\}} (\bar{v}_1^K, e_{J+1}^K), \quad (2.32)$$

$$\left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,J\}} (\bar{v}_{I+1}^K, e_{J+1}^K) = \left. \frac{\partial^2 p}{\partial \bar{v} \partial e} \right|_{\{I,J\}} (\bar{v}_{I+1}^K, e_{J+1}^K). \quad (2.33)$$

The linear system of Eqs. (2.13 - 2.33) is solved in order to obtain the $9IJ$ spline coefficients a_{ijkl} . The resulting spline function and its first derivatives are continuous over its full domain of definition. A more general description of the SBTL method is given in [1, 11, 12].

3.5 Calculation of inverse functions using the example of $e(p, v)$

Analogously to the one-dimensional inverse function discussed in Section 3.3, a numerically consistent inverse spline function for $e(p, v)$ is obtained from the spline function for $p(v, e)$. The bi-quadratic polynomial spline function given in Eq. (1.6) is solved in terms of the specific internal energy e with

$$e_{\{i,j\}}^{\text{INV}}(p, \bar{v}) = \frac{\left(-B_{ij} \pm \sqrt{B_{ij}^2 - 4A_{ij}C_{ij}(p)} \right)}{2A_{ij}} + e_j, \quad (2.34)$$

where

$$A_{ij} = a_{ij13} + \Delta \bar{v}_i (a_{ij23} + a_{ij33} \Delta \bar{v}_i),$$

$$B_{ij} = a_{ij12} + \Delta\bar{v}_i (a_{ij22} + a_{ij32}\Delta\bar{v}_i), \text{ and}$$

$$C_{ij}(p) = a_{ij11} + \Delta\bar{v}_i (a_{ij21} + a_{ij31}\Delta\bar{v}_i) - p.$$

For monotonic functions $p_{\{i,j\}}(e)_{\bar{v}}$ in cell $\{i,j\}$, the sign (\pm) in Eq. (2.34) equals $\text{sgn}(B_{ij})$. In order to calculate $e(p,v)$ from Eq. (2.34), the corresponding cell $\{i,j\}$, where the given state point (p,v) is located, must be determined. For this purpose, the auxiliary spline function $e^{\text{AUX}}(p,v)$ and Eqs. (2.10, 2.11) are used. Then, the inverse spline polynomial $e_{\{i,j\}}^{\text{INV}}(p,\bar{v})$, Eq. (2.34), is evaluated. The result must fulfil the condition described by Eq. (2.11), otherwise, the index j needs to be incremented or decremented and the calculation repeated.

For the spline function $p(v,e)$ for the liquid region L a scaling function, Eq. (2.1), with the dependent variable e of the inverse spline function $e(p,v)$ is applied. In this case, an analytic solution for the inverse spline function cannot be provided. Fast iterative procedures to solve this problem are given in [12]. Due to the volume anomaly of water at low temperatures, the property function $e(p,v)$ is non-monotonic in this range of state and two possible solutions must be distinguished as discussed in [1, 11, 12]. Inverse functions of input variables that are both neither v nor e , such as (p,h) , (p,s) , and (h,s) need to be calculated from the spline functions of (v,e) by iteration. For this purpose, fast algorithms for calculating the desired variables by iteration from a non-linear system of equations are used as described in [12]. Auxiliary spline functions have been generated to calculate the required initial values for v and e .

3.6 SBTL functions for the single-phase region

For each of the two single-phase regions, L and G, spline functions for the calculation of p , T , s , w , η , $\lambda = f(v,e)$ have been generated. The specific enthalpy h and the specific Gibbs free energy g are calculated from $h=e+pv$ and $g=h-Ts$. Since the computation of $g(v,e)$ involves the evaluation of $p(v,e)$, $T(v,e)$, and $s(v,e)$, additional spline functions are provided for even faster calculations of $g(v,e)$. The isobaric and isochoric heat capacities are calculated from $p(v,e)$ and $T(v,e)$ according to their definitions

$$c_p = \left(\frac{\partial h}{\partial T} \right)_p = \frac{\left(\frac{\partial p}{\partial v} \right)_e - p \left(\frac{\partial p}{\partial e} \right)_v}{\left(\frac{\partial T}{\partial e} \right)_v \left(\frac{\partial p}{\partial v} \right)_e - \left(\frac{\partial T}{\partial v} \right)_e \left(\frac{\partial p}{\partial e} \right)_v} \text{ and} \quad c_v = \left(\frac{\partial e}{\partial T} \right)_v.$$

Since c_p and c_v contain the first derivatives of the bi-quadratic spline functions for $p(v,e)$ and $T(v,e)$, the first derivatives of c_p and c_v are discontinuous. In many process simulations, these derivatives are not required. For all other cases, additional spline functions for the calculation of $c_p(v,e)$ and $c_v(v,e)$ have been generated.

The correlating equations for the dynamic viscosity η [14] and the thermal conductivity λ [15] contain critical enhancement terms to describe the behavior of these properties in the critical region. The critical enhancement terms depend on the derivative $(\partial v / \partial p)_T$, which is infinite along the spinodals. This causes numerical difficulties and therefore, the critical enhancement terms were omitted for the generation of the $\eta(v,e)$ and $\lambda(v,e)$ spline functions. For the dynamic viscosity η the critical enhancement is only significant in a very small region around the critical point and the omission of the critical enhancement is recommended in [14]

to simplify the calculation for industrial use. For the thermal conductivity λ , the critical enhancement is significant in a larger range around the critical point as discussed in [15].

Table 1: Dimensions of the grid of nodes of each (v,e) spline function for the liquid region L based on IAPWS-95 and the IAPWS releases on viscosity and thermal conductivity [14, 15]

Spline function	v [m ³ /kg]	e [kJ/kg]
	$\bar{v}(v,u)$ (see Eq. (2.1))	-
	$\begin{bmatrix} \bar{v}_{\min} \\ \dots \\ \dots \\ \bar{v}_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix}$	$\begin{bmatrix} e_{\min} \\ \dots \\ \dots \\ e_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix}$
$p^L(v,e)$	$\begin{bmatrix} 1 \\ 20 \\ 40 \\ 95 \\ 100 \end{bmatrix} \begin{bmatrix} 50 \\ 150 \\ 75 \\ 75 \end{bmatrix}$	$\begin{bmatrix} -8 \\ 250 \\ 2040.02 \end{bmatrix} \begin{bmatrix} 350 \\ 225 \end{bmatrix}$
$T^L(v,e)$	$\begin{bmatrix} 1 \\ 100 \end{bmatrix} [100]$	$\begin{bmatrix} -8 \\ 2040.02 \end{bmatrix} [200]$
$s^L(v,e)$	$\begin{bmatrix} 1 \\ 100 \end{bmatrix} [100]$	$\begin{bmatrix} -8 \\ 10 \\ 2040.02 \end{bmatrix} \begin{bmatrix} 10 \\ 200 \end{bmatrix}$
$w^L(v,e)$	$\begin{bmatrix} 1 \\ 95 \\ 100 \end{bmatrix} \begin{bmatrix} 100 \\ 50 \end{bmatrix}$	$\begin{bmatrix} -8 \\ 10 \\ 1750 \\ 2040.02 \end{bmatrix} \begin{bmatrix} 10 \\ 200 \\ 75 \end{bmatrix}$
$\eta^L(v,e)$	$\begin{bmatrix} 1 \\ 100 \end{bmatrix} [100]$	$\begin{bmatrix} -8 \\ 300 \\ 2040.02 \end{bmatrix} \begin{bmatrix} 75 \\ 150 \end{bmatrix}$
$\lambda^L(v,e)$	$\begin{bmatrix} 1 \\ 100 \end{bmatrix} [100]$	$\begin{bmatrix} -8 \\ 10 \\ 1750 \\ 2040.02 \end{bmatrix} \begin{bmatrix} 10 \\ 50 \\ 50 \end{bmatrix}$

Table 2: Dimensions of the grid of nodes of each (v,e) spline function for the gas region G based on IAPWS-95 and the IAPWS releases on viscosity and thermal conductivity [14, 15]

Spline function	v [m ³ /kg]	e [kJ/kg]
	$\bar{v}(v) = \ln(v)$	-
	$\begin{bmatrix} \bar{v}_{\min} \\ \dots \\ \dots \\ \bar{v}_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix}$	$\begin{bmatrix} e_{\min} \\ \dots \\ \dots \\ e_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix}$
$p^G(v,e)$ $T^G(v,e)$ $s^G(v,e)$ $w^G(v,e)$ $\eta^G(v,e)$ $\lambda^G(v,e)$	$\begin{bmatrix} \bar{v}(1.6 \times 10^{-3}) \\ \bar{v}(8 \times 10^{-3}) \\ \bar{v}(1189.01) \end{bmatrix} \begin{bmatrix} 100 \\ 200 \end{bmatrix}$	$\begin{bmatrix} 1845 \\ 2650 \\ 4085.58 \end{bmatrix} \begin{bmatrix} 100 \\ 100 \end{bmatrix}$

The grid dimensions of each (v,e) spline function are given in Tables 1 and 2. In the liquid region L, a scaling transformation for the specific volume v of the form $\bar{v}(v,e)$, Eq. (2.1), is applied. Thus, the shape of the grid of nodes corresponds to the shape of the liquid region L (see Figs. 1, 5, and 6). In the gas region G, the specific volume v is transformed as $\bar{v} = \ln(v)$. For piecewise equidistant nodes, the domain of the considered variable $\bar{v}_{\min} \leq \bar{v} \leq \bar{v}_{\max}$ or $e_{\min} \leq e \leq e_{\max}$ is subdivided in several intervals with equidistant nodes. In the tables below, this is described with

$$\begin{bmatrix} \bar{v}_{\min} \\ \dots \\ \dots \\ \bar{v}_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix} \text{ and } \begin{bmatrix} e_{\min} \\ \dots \\ \dots \\ e_{\max} \end{bmatrix} \begin{bmatrix} \text{nodes} \\ \dots \\ \dots \\ \text{nodes} \end{bmatrix},$$

where the boundaries of the intervals are given in the column on the left and the number of equidistant nodes between them is given in the column on the right. Nodes outside the range of validity are extrapolated appropriately. Inverse functions for calculations from (p,T) , (p,v) , (p,h) , (p,s) , and (h,s) have been implemented as described in Section 3.5. For each input variable combination, analytical derivatives are provided.

3.7 Property calculations in the two-phase region

The saturation states could be calculated from the Maxwell criterion, i.e., equal pressures and specific Gibbs free energies at constant temperature for both phases; but for the sake of simplicity, a spline function for $T_s(p)$ is used instead. For property functions of (p,v) , (p,h) , and (p,s) , the properties at saturation from can be calculated from the inverse functions, such

as $v' = v^{L,INV}(p, T = T_s(p))$ and $v'' = v^{G,INV}(p, T = T_s(p))$. In order to calculate the properties at saturation from (v, e) , the following set of equations $\mathbf{F}(\mathbf{X})$, Eqs. (3.1 - 3.5),

$$\mathbf{F}_1(\mathbf{X}) = 0 = p^L(v', e') - p_s, \quad (3.1)$$

$$\mathbf{F}_2(\mathbf{X}) = 0 = p^G(v'', e'') - p_s, \quad (3.2)$$

$$\mathbf{F}_3(\mathbf{X}) = 0 = T^L(v', e') - T_s(p_s), \quad (3.3)$$

$$\mathbf{F}_4(\mathbf{X}) = 0 = T^G(v'', e'') - T_s(p_s), \text{ and} \quad (3.4)$$

$$\mathbf{F}_5(\mathbf{X}) = 0 = \frac{v - v'}{v'' - v'} - \frac{e - e'}{e'' - e'} \quad (3.5)$$

is solved for the vector of unknowns $\mathbf{X} = (p_s, v', v'', e', e'')^T$. This is done through the use of Newton's method for non-linear systems of equations by solving

$$\mathbf{J}(\mathbf{X}_k) \Delta \mathbf{X}_k = \mathbf{F}(\mathbf{X}_k) \text{ and} \quad (3.6)$$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \Delta \mathbf{X}_k \quad (3.7)$$

in each iteration step k until convergence is reached. The Jacobian matrix $\mathbf{J}(\mathbf{X})$ is given analytically as shown in [1, 11, 12]. Initial values of the unknown variables are calculated from auxiliary spline functions for $p_s(v, e)$ and for $v'(T)$, $v''(T)$, $e'(T)$, and $e''(T)$. The vapor mass fraction x is calculated from

$$x = (v - v') / (v'' - v') \text{ or } x = (e - e') / (e'' - e'). \quad (3.8)$$

Then, the other mass-averaged properties can be calculated from the properties at saturation, as for example the specific entropy s from $s' = z^L(v', e')$, $s'' = z^G(v'', e'')$, and

$$s(v, e) = s' + x(s'' - s'). \quad (3.9)$$

Additionally, algorithms for calculating the properties in the two-phase region have been implemented for (p, x) , (T, x) , (p, v) , (p, h) , (p, s) , and (h, s) inputs. For each input variable combination, analytical derivatives are provided.

3.8 Deviations of the SBTL functions from IAPWS-95

For the development of SBTL functions for the liquid region L and the gas region G, the permissible deviations from IAPWS-95 as well as from the IAPWS formulations for viscosity and thermal conductivity shown in Table 3 have been set. The SBTL functions represent the IAPWS formulations within these permissible deviations. The deviations in pressure $p(v, e)$, temperature $T(v, e)$, and specific entropy $s(v, e)$ from IAPWS-95 in the liquid region L and the gas region G are depicted in Figs. 9, 10, and 11.

Table 3: Permissible deviations for the development of SBTL functions from IAPWS-95 as well as the IAPWS formulations for viscosity and thermal conductivity [14, 15]

SBTL function	Liquid region L	Gas region G
$p(v,e)$	$p \leq 2.5 \text{ MPa}$ $p > 2.5 \text{ MPa}$	
$T(v,e)$		
$s(v,e)$		
$w(v,e)$		
$\eta(v,e)^a$		
$\lambda(v,e)^a$		

^a The critical enhancement of dynamic viscosity and thermal conductivity has been omitted (see Section 3.6 for details.)

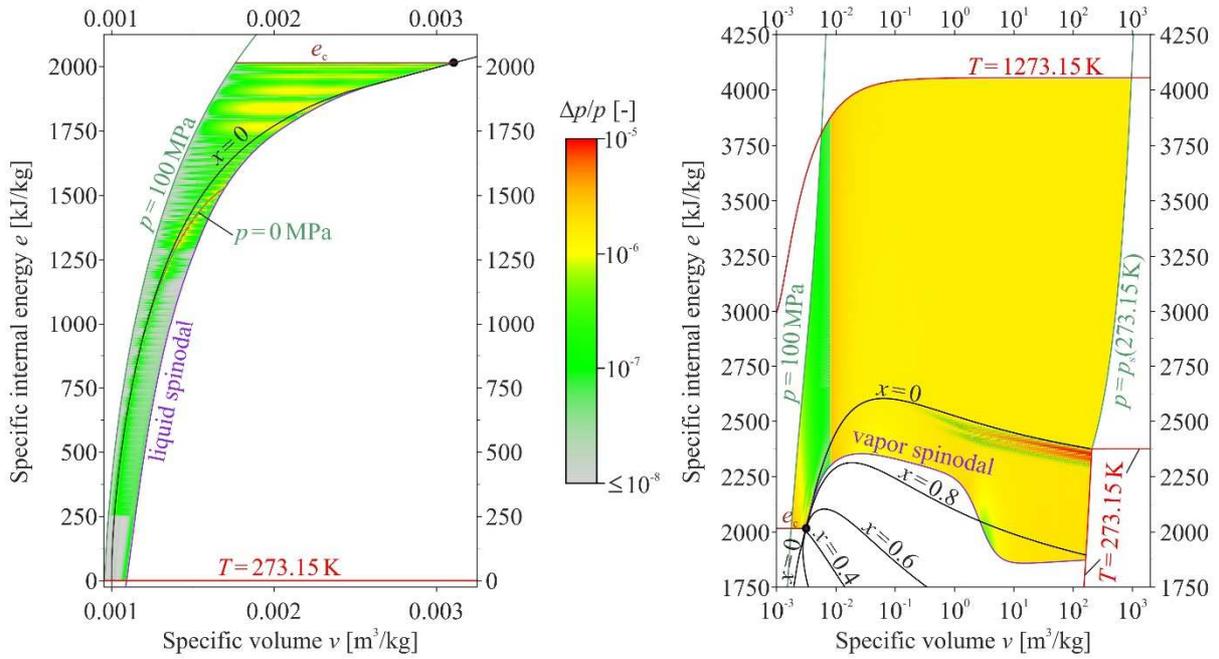


Figure 9: Deviations in $p(v,e)$ from IAPWS-95 in the liquid region L and the gas region G.

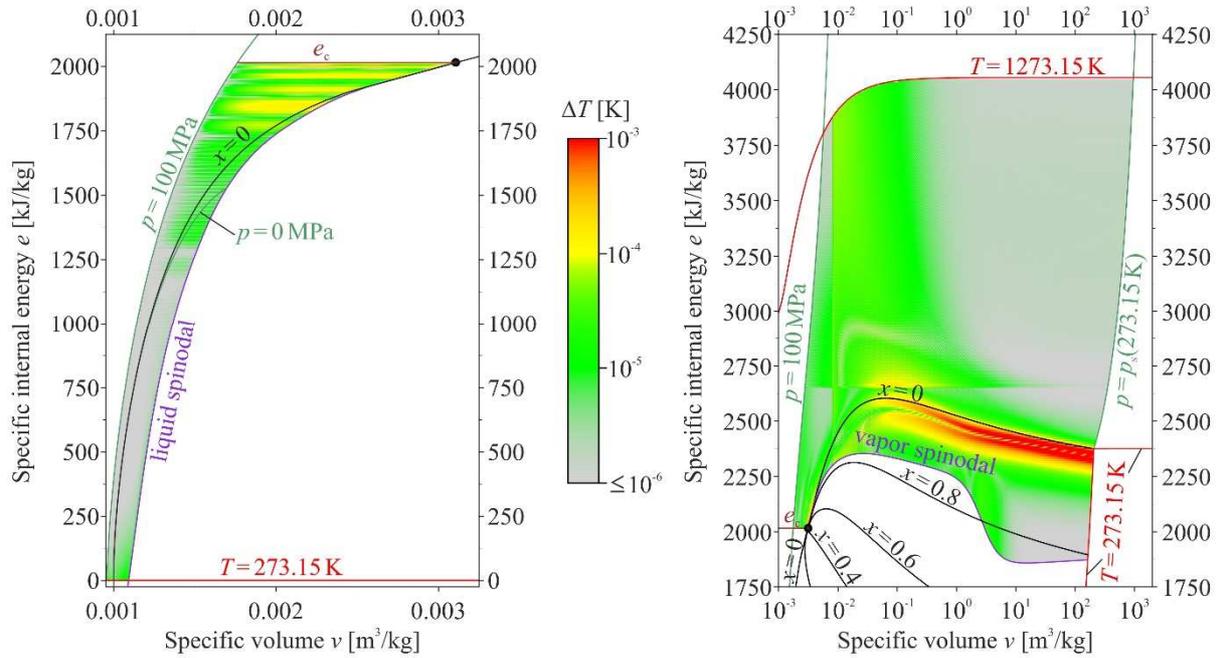


Figure 10: Deviations in $T(v,e)$ from IAPWS-95 in the liquid region L and the gas region G.

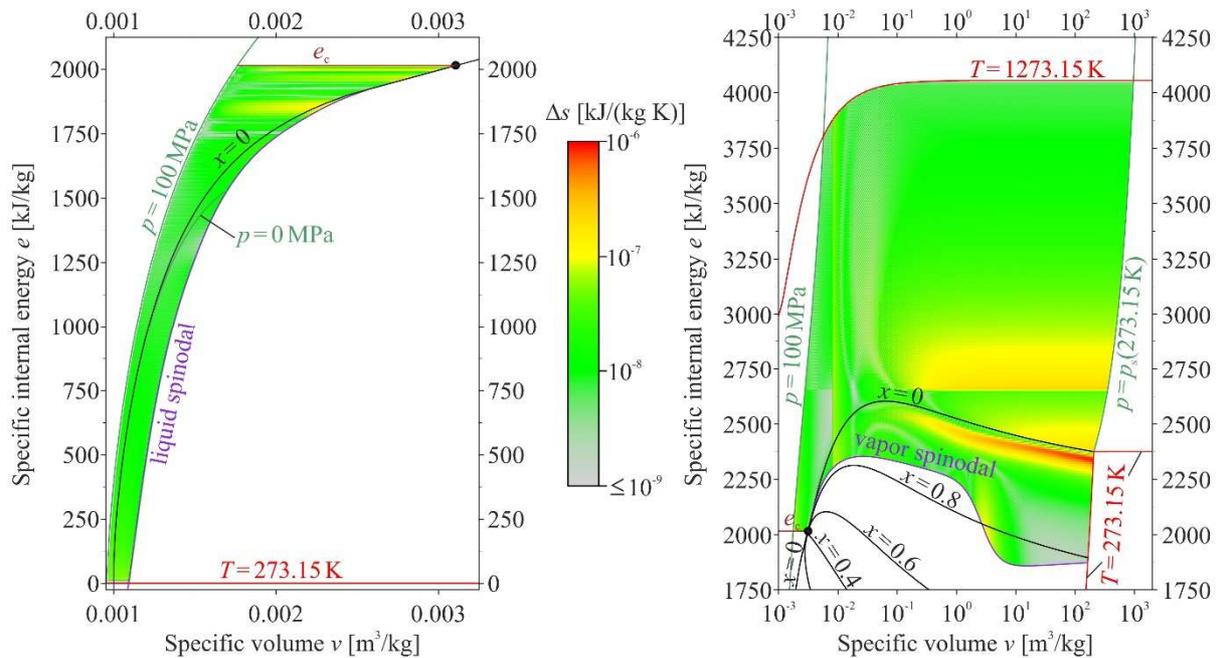


Figure 11: Deviations in $s(v,e)$ from IAPWS-95 in the liquid region L and the gas region G.

3.9 Computing-times comparisons

The calculation of property functions from IAPWS-95 is computationally intensive. Therefore, the computing times of the spline functions described in Sections 3.6 and 3.7 have been compared with those of the computationally more efficient IAPWS-IF97 formulation, which is used in many industrial applications. To illustrate the difference in computing speed between IAPWS-95 and IAPWS-IF97, the computing times of the spline functions for $p(v,e)$ and $T(v,e)$ have also been compared with IAPWS-95.

The Computing-Time Ratio (*CTR*) is:

$$CTR = \frac{\text{Computing time of the calculation from IAPWS-IF97 (IAPWS-95)}}{\text{Computing time of the calculation from the SBTL algorithms}}$$

IAPWS-IF97 property functions were computed from the Extended IAPWS-IF97 Steam Tables software [16]. Since the region definitions of the SBTL functions are different from the regions of IAPWS-IF97, the computing times of both formulations include the determination of the region that corresponds to the given state point. The IAPWS-95 property functions were computed from the internal routines of REFPROP [17], where the phase (liquid or vapor) is known and no phase tests are performed. None of the implementations of IAPWS-IF97, IAPWS-95, or the SBTL method takes advantage of information from previously calculated state points. The computing times were measured by means of software similar to NIFBENCH [5] with 100,000 randomly distributed state points in the corresponding region. All algorithms have been compiled into single-threaded software with the Intel Composer 2011 with default options. The tests were carried out on a Windows 8 computer equipped with an Intel Core i7-4500U CPU with 2.39 GHz and 8 GB RAM. The results of the computing-time comparisons are summarized in Table 4.

Table 4: Computing-time ratios (*CTR*) of SBTL property functions in comparison to the iterative calculations from IAPWS-IF97 and from IAPWS-95 (in parentheses)

SBTL function	IAPWS-IF97 Region				
	1	2	3	4	5
$p(v,e)$	130 (243 ^a)	271 (434 ^a)	161	19.6	470
$T(v,e)$	161 (251 ^a)	250 (410 ^a)	158	20.6	442
$s(v,e)$	164	261	160	17.8	449
$e(p,v)$	2.0	6.4	2.8	5.6	3.2
$T(p,h)^a$	2.9 ($\approx 15\,000$)	4.7 (6760)	3.0	4.4	26.5
$v(p,h)^a$	3.8 ($\approx 14\,500$)	6.1 (6900)	5.1	2.6	25.2

^a Calculated from explicit spline functions rather than from inverse spline functions.

4 SBTL property functions for other fluids

In order to apply the SBTL method to property functions for any fluid, the software FluidSplines [12, 18] has been developed. The underlying fluid properties are provided from external databases such as the property libraries from the Zittau/Görlitz University of Applied Sciences or REFPROP [17] from NIST. FluidSplines implements all the features of the SBTL method and assists the user in generating spline functions and inverse spline functions for a given range of validity with a user-specified agreement with the underlying property formulations.

5 Application of the SBTL method in RELAP-7

RELAP-7 (Reactor Excursion and Leak Analysis Program) [2] is the next generation nuclear reactor system safety analysis code currently being developed at Idaho National Laboratory (INL). The code is based on the INL's scientific software development framework MOOSE (Multi-physics Object Oriented Simulation Environment) [19]. By using the MOOSE development framework, RELAP-7 can be tightly (implicitly) coupled to over 20 different MOOSE based applications ranging from 3-D transient neutron transport, to detailed 3-D transient fuel performance analysis, to long-term material aging, thus leveraging with capabilities developed through other programs.

The primary basis of the RELAP-7 governing theory includes, at a system level, nonequilibrium two-phase flow, reactor core heat transfer, and reactor kinetics models. Because of the broad spectrum of phenomena occurring in light water nuclear reactor coolant flows (boiling, flashing, and bubble collapse, choking, blowdown, condensation, wave propagation, large density variations, etc.) it is imperative that models accurately describe compressible multiphase flows exhibiting metastable states, with multiple velocities, and that the models be well-posed (in the sense of Hadamard) and unconditionally hyperbolic. A detailed summary of model requirements may be found in [20]. It is known that van der Waals or cubic equations of state used with Euler or Navier-Stokes equations correspond to ill-posed mixture models because the square of the sound speed becomes negative in the spinodal zone. To avoid this difficulty we utilize a model in RELAP-7 that considers phase change as a kinetic transformation rather than as a thermodynamic one. In this model each phase retains its own pure substance equation of state. Thus, each phase has its own dynamics and thermodynamics (velocity, density, temperature, pressure, etc.), and the modeling of mass transfer is a relaxation towards equilibrium by the kinetic process. With such a kinetic representation, metastable states are present, the mixture sound speed is always well defined, and hyperbolicity is preserved during interphase mass transfer. Furthermore, when equilibrium is reached, the usual properties of the phase diagram, such as latent heat of vaporization and saturation temperature, are recovered.

To meet this criterion, we have adopted the 7-equation model [20-24] for fully nonequilibrium, fully compressible, two-phase flows. This equation system meets our requirements, being both hyperbolic and well-posed, and it has an intuitively pleasing set of genuinely nonlinear and linear degenerate eigenvalues (wave speeds). This same 7-equation two-phase flow model is also being utilized to build the next generation 3-D high-resolution, multi-scale two-phase solver to seamlessly couple with the RELAP-7 systems analysis code.

The one-dimensional form, with variable cross-section flow area, of the fully nonequilibrium, 7-equation two-phase flow model used by RELAP-7 is described by the balance of mass, momentum, and total energy for each phase (liquid and vapor) plus a volume fraction evolution equation for each phase (except as noted below for two phases), respectively, as

liquid phase:

$$\frac{\partial \alpha_l \rho_l A}{\partial t} + \frac{\partial \alpha_l \rho_l u_l A}{\partial x} = -\Gamma_{int, g} A_{int} - \Gamma_{wall, g} \quad (4.1)$$

$$\begin{aligned}
\frac{\partial \alpha_l \rho_l u_l A}{\partial t} + \frac{\partial \alpha_l A (\rho_l u_l^2 + p_l)}{\partial x} &= p_{int} A \frac{\partial \alpha_l}{\partial x} + p_l \alpha_l \frac{\partial A}{\partial x} + A \lambda (u_g - u_l) \\
&\quad - \Gamma_{int, g} A_{int} u_{int} A - \Gamma_{wall, g} u_{int} \\
&\quad - F_{wall \text{ friction}, l} - F_{int \text{ friction}, g} + \alpha_l \rho_l A \vec{g} \cdot \hat{n}_{axis}
\end{aligned} \tag{4.2}$$

$$\begin{aligned}
\frac{\partial \alpha_l \rho_l E_l A}{\partial t} + \frac{\partial \alpha_l u_l A (\rho_l E_l + p_l)}{\partial x} &= p_{int} u_{int} A \frac{\partial \alpha_l}{\partial x} \\
&\quad - \bar{p}_{int} A \mu (p_l - p_g) + \bar{u}_{int} A \lambda (u_g - u_l) \\
&\quad + \Gamma_{int, g} A_{int} \left(\frac{p_{int}}{\rho_{int}} - H_{l, int} \right) A \\
&\quad + A_{int} h_{conv, l} (T_{int} - T_l) A \\
&\quad + Q_{wall, l, conv} \\
&\quad - \Gamma_{wall, g} \left(-\frac{p_{int}}{\rho_{int}} + h_{g, int} + \frac{1}{2} u_{int}^2 \right) \\
&\quad + \alpha_l \rho_l u_l A \vec{g} \cdot \hat{n}_{axis}
\end{aligned} \tag{4.3}$$

$$\frac{\partial \alpha_l A}{\partial t} + u_{int} A \frac{\partial \alpha_l}{\partial x} = A \mu (p_l - p_g) - \frac{\Gamma_{int, g} A_{int} A}{\rho_{int}} - \frac{\Gamma_{wall, g}}{\rho_{int}} \tag{4.4}$$

vapor phase:

$$\frac{\partial \alpha_g \rho_g A}{\partial t} + \frac{\partial \alpha_g \rho_g u_g A}{\partial x} = \Gamma_{int, g} A_{int} A + \Gamma_{wall, g} \tag{4.5}$$

$$\begin{aligned}
\frac{\partial \alpha_g \rho_g u_g A}{\partial t} + \frac{\partial \alpha_g A (\rho_g u_g^2 + p_g)}{\partial x} &= p_{int} A \frac{\partial \alpha_g}{\partial x} + p_g \alpha_g \frac{\partial A}{\partial x} + A \lambda (u_l - u_g) \\
&\quad + \Gamma_{int, g} A_{int} u_{int} A + \Gamma_{wall, g} u_{int} \\
&\quad - F_{wall \text{ friction}, g} - F_{int \text{ friction}, g} + \alpha_g \rho_g A \vec{g} \cdot \hat{n}_{axis}
\end{aligned} \tag{4.6}$$

$$\begin{aligned}
\frac{\partial \alpha_g \rho_g E_g A}{\partial t} + \frac{\partial \alpha_g u_g A (\rho_g E_g + p_g)}{\partial x} &= p_{int} u_{int} A \frac{\partial \alpha_g}{\partial x} \\
&\quad - \bar{p}_{int} A \mu (p_g - p_l) + \bar{u}_{int} A \lambda (u_l - u_g) \\
&\quad - \Gamma_{int, g} A_{int} \left(\frac{p_{int}}{\rho_{int}} - H_{g, int} \right) A \\
&\quad + A_{int} h_{conv, g} (T_{int} - T_g) A \\
&\quad + Q_{wall, g, conv} \\
&\quad + \Gamma_{wall, g} \left(-\frac{p_{int}}{\rho_{int}} + h_{g, int} + \frac{1}{2} u_{int}^2 \right) \\
&\quad + \alpha_g \rho_g u_g A \vec{g} \cdot \hat{n}_{axis}
\end{aligned} \tag{4.7}$$

$$\frac{\partial \alpha_g A}{\partial t} + u_{int} A \frac{\partial \alpha_g}{\partial x} = A \mu (p_g - p_l) + \frac{\Gamma_{int, g} A_{int} A}{\rho_{int}} + \frac{\Gamma_{wall, g}}{\rho_{int}} \tag{4.8}$$

For two phases, Eq. (4.8) is conveniently replaced by the (algebraic) saturation constraint

$$\alpha_l + \alpha_g = 1. \quad (4.9)$$

In the balance equations above, the *interfacial variables* are

$$p_{int} = \bar{p}_{int} + \frac{Z_l Z_g}{Z_l + Z_g} \operatorname{sgn}\left(\frac{\partial \alpha_l}{\partial x}\right) (u_g - u_l) \quad (4.10)$$

$$\bar{p}_{int} = \frac{Z_l p_g + Z_g p_l}{Z_l + Z_g} \quad (4.11)$$

$$u_{int} = \bar{u}_{int} + \frac{Z_l Z_g}{Z_l + Z_g} \operatorname{sgn}\left(\frac{\partial \alpha_l}{\partial x}\right) \frac{p_g - p_l}{Z_l + Z_g} \quad (4.12)$$

$$\bar{u}_{int} = \frac{Z_l u_l + Z_g u_g}{Z_l + Z_g} \quad (4.13)$$

$$Z_k = \rho_k c_k \quad (k = l, g) \quad (4.14)$$

$$\rho_{int} = \rho_{l, sat}(\bar{p}_{int}) \quad (4.15)$$

$$H_{k, int} = h_{k, int} + \frac{1}{2} u_{int}^2 \quad (k = l, g) \quad (4.16)$$

with the *mechanical relaxation coefficients* given by

$$\lambda = \frac{1}{2} \mu Z_l Z_g \quad (4.17)$$

$$\mu = \frac{A_{int}}{Z_l + Z_g} \quad (4.18)$$

and the *interphase mass transfer* given by

$$\Gamma_{int, g} = \frac{h_{conv, l}(T_l - T_{int}) + h_{conv, g}(T_g - T_{int})}{h_{g, int} - h_{l, int}} = \frac{h_{conv, l}(T_l - T_{int}) + h_{conv, g}(T_g - T_{int})}{L_v(T_{int})} \quad (4.19)$$

where

$$T_{int} = T_{sat}(\bar{p}_{int}) \quad (4.20)$$

$$h_{k, int} = h_{k, sat}(T_{int}) \quad (k = l, g). \quad (4.21)$$

In the equations above, the notations are classical from the two-phase flow literature.

If full *mechanical relaxation* (in which phasic pressures and velocities are relaxed to single values, see for example [25]) along with full *thermodynamic relaxation* (in which phasic temperatures and Gibbs energies are relaxed to single values) occurs, the 7-equation model above dynamically reduces to the 3-equation Homogeneous Equilibrium Model (HEM) model. Formally, the three governing balance equations of the HEM model are identical to the Euler equations; however, the density is an equilibrium volume fraction weighted mixture value, the specific internal energy is an equilibrium mass fraction weighted mixture value, the pressure and temperature are thermodynamic equilibrium values, and the velocity is the mixture barycentric value. Though the 3-equation HEM model is the simplest and oldest two-

phase model it has many limitations and some of its properties, e.g. effective sound speed, are more difficult to compute, and may even exhibit discontinuities in transitions from single- to two-phases. For applications where the HEM representation is physically appropriate, it is more economical to begin with the 3-equation model, rather than carry the additional expense of the relaxed 7-equation model. The 3-equation HEM model is included also as a selectable model in RELAP-7.

For the accurate simulation of two-phase flows with RELAP-7 the equation systems must be closed (partially) with accurate thermodynamic equations of state to obtain the properties of steam and water. Moreover, for CPU-intensive numerical simulations with this code, thermodynamic and transport properties of steam and water are calculated extremely often. Examination of the governing partial differential equations of the 7-equation two-phase model shows that their dependent variables are mass-, momentum-, and total energy-densities. Thus the thermodynamics variables resulting from their solution are phasic specific volume and specific internal energy (v_k, e_k)

$$v_k = \frac{1}{\rho_k} = \frac{\alpha_k}{(\alpha\rho)_k} \quad (k=l, g) \quad (4.22)$$

$$e_k = \frac{(\alpha\rho E)_k}{(\alpha\rho)_k} - \frac{1}{2} \frac{(\alpha\rho u)_k^2}{(\alpha\rho)_k^2} \quad (k=l, g) \quad (4.23)$$

or, in the case of the HEM two-phase model, the corresponding solution dependent variables are equilibrium mixture values (v, e). Determining properties as a function of (v_k, e_k) or (v, e) from an accurate equation of state such as IAPWS-95 would normally require a functional inversion, because these are not the dependencies upon which it is constructed, based upon an iterative numerical solution that is very time-consuming and not computationally efficient. In addition, for initial condition specification and boundary condition calculations, other thermodynamic dependencies are needed, such as $e_k(v_k, p_k)$, $e_k(s_k, p_k)$, $v_k(s_k, p_k)$, and $p_k(h_k, s_k)$. To provide fast and accurate property calculation algorithms, RELAP-7 was modified to employ the Spline-Based Table Look-up (SBTL) method based on IAPWS-95, as described above. Because each phase of the 7-equation two-phase model treats only that phase, which can be in either the stable or metastable (i.e. in a state between the saturation line and the appropriate spinodal line) state, the IAPWS-95/SBTL package was modified to accommodate this need. For the HEM two-phase model the IAPWS-95/SBTL package produces only equilibrium properties. The utilization of the SBTL method has enabled RELAP-7 to eliminate the approximate stiffened gas analytical equations of state and to complete the major hurdle of incorporating the numerically efficient calculation of accurate water and steam thermodynamic properties [2].

To fully close the 7-equation two-phase model, additional relations must be supplied which are very significantly dependent upon the flow topology (or *flow regime*) such as relationships between the fluid phases and the wall (friction $F_{wall, k}$, heat-transfer $Q_{wall, k, conv}$, and mass-transfer $\pm\Gamma_{wall, g}$) and relationships describing the bulk interactions between the phases (interphase mass-transfer $\pm\Gamma_{int, g}$, momentum-transfer, and energy-transfer) which depend strongly upon the interfacial area concentration and distribution. Many of these flow topology-dependent closure correlations have been extensively investigated and partially

validated in existing system analysis codes. Such closures are currently being implemented into RELAP-7 and are partially described in [2].

Because these complex flow topology-dependent closures are a work in progress, it will not be reported herein. However, a unique advantage of the 7-equation two-phase model is its ability to execute in a completely nonequilibrium manner, i.e. with absolutely no interaction between the phases other than sharing the same flow channel, which will effectively nullify the important physical effects due to flow topology [22]. For example, let us consider a two-phase flow of water and its vapor through a converging-diverging nozzle with cosine variation of the cross-sectional area such that the inlet cross-section has the same area as that of the outlet, *cosine-nozzle*. If (1) the interphase heat and mass transfers are turned off, (2) the pressure and velocity relaxation coefficients are set also to zero, and (3) the initial volume fraction is spatially constant with (4) the inlet boundary volume fraction set to match this constant value for all times, then the two-phase flow solution of the 7-equation model should identically match that of two, single-phase flows (water and its vapor) each flowing through the same nozzle. We consider the symmetrical cosine-nozzle to be of length 1.0 meter, with $A_{throat}/A_{exit}=0.33$, and being spatially discretized with 1000 continuous, linear finite elements. The nozzle problem is executed as a transient problem using second order backward difference (implicit) time integration and run to steady state at approximately 1.1 s. Both phases have the same inlet *stagnation* boundary condition specified at approximately the saturation condition, $p_0=1.0\cdot 10^6 Pa$ and $T_0=453 K$ at $x=0.0$ for all time. The inlet liquid volume fraction $\alpha_l=0.5$ is specified at $x=0.0$ for all time. The nozzle's liquid volume fraction spatial distribution is specified as $\alpha_l=0.5$ for $0.0\leq x\leq 1.0$ at time $t=0.0$. The nozzle's remaining initial conditions are $p=1.0\cdot 10^6 Pa$, $T=453 K$, and velocity $v=0.0$ for $0.0\leq x\leq 1.0$ (stationary saturation condition) at time $t=0.0$. The nozzle's outlet boundary condition is specified *static* pressure for both phases of $p_b=0.5\cdot 10^6 Pa$ at $x=1.0$ for all time. Steady state solutions for this nonequilibrium problem are shown in Figures 12 and 13 where it is seen that, because phase change and interphase heat transfer are not allowed and because phasic pressures and velocities are not allowed to equilibrate, both compressible phases must necessarily acquire states significantly into the *metastable regions*. A few observations are apparent. The vapor solution is the classical compressible solution with a standing shock wave in the diverging portion of the nozzle at approximately 0.82 meters. Examining the liquid phase pressure, the liquid solution might at first appear to be classical *incompressible* venturi flow. However, upon further examining the liquid phase density and temperature profiles it is revealed that the liquid phase solution indeed follows the classical compressible flow solution for fully subsonic flow (shock free). This is as it should rightly be, because RELAP-7 treats both the liquid and vapor phases as being compressible. For this low-speed, lossless flow the computed static pressure at the inlet should exactly match the specified static pressure set for the outlet.

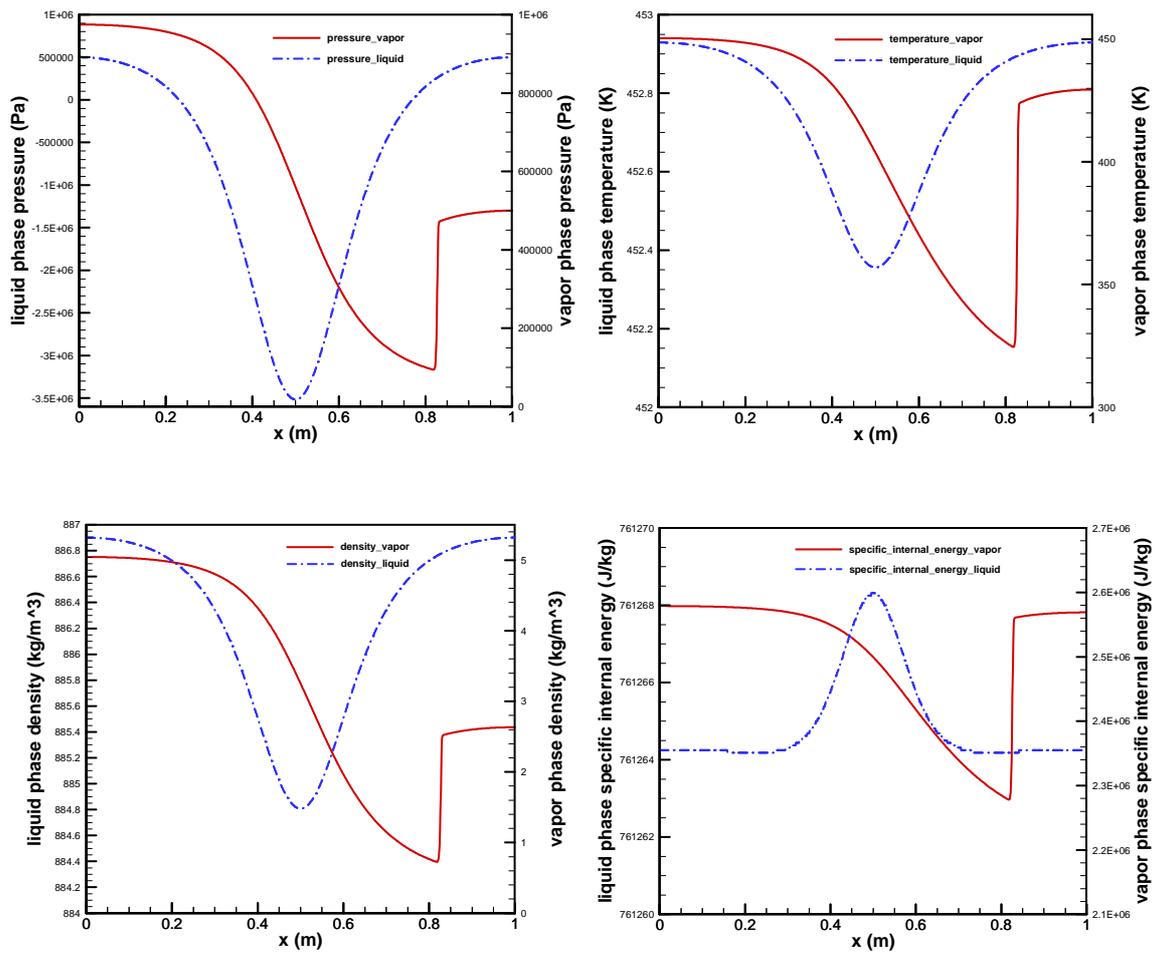


Figure 12: Nozzle phasic thermodynamic property at steady state for completely nonequilibrium flow.

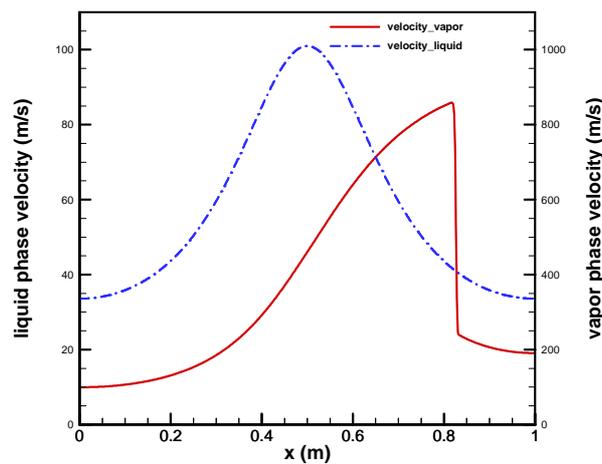


Figure 13: Nozzle phasic velocity distribution at steady state for completely nonequilibrium flow.

6 Conclusions and outlook

The new “IAPWS Guideline on the Fast Calculation of Steam and Water Properties with the Spline-Based Table Look-Up Method (SBTL)” has been applied to the scientific formulation for water and steam IAPWS-95 and the latest IAPWS formulations for viscosity and thermal conductivity. The newly generated SBTL functions include the metastable liquid and the metastable vapor regions and are applicable for pressures up to 100 MPa and temperatures up to 1273.15 K. Depending on the considered property and the range of state, the deviations of the SBTL functions from their underlying property formulations are 1-100 ppm. The property functions of specific volume and specific internal energy (v, e), as used in RELAP-7, are more than 400 times faster than the iterative calculation from IAPWS-95. Moreover, fast inverse functions of pressure and temperature (p, T), pressure and specific volume (p, v), pressure and specific enthalpy (p, h), pressure and specific entropy (p, s), and specific enthalpy and specific entropy (h, s) have been developed. The SBTL functions and their inverse functions are numerically consistent with each other and continuously differentiable once.

The SBTL functions have been implemented into the nuclear reactor system safety analysis code RELAP-7 to consider the real fluid behavior of water and steam and to enable the utilization of a novel 7-equation two-phase flow model. Prior to the incorporation of the SBTL method in RELAP-7, a stiffened gas equation of state was used for each phase with appropriate modification to link the reference states [26]. This was useful for achievement of rapid development progress, but for use with real nuclear power plants, accurate water and steam properties are essential, not only for equilibrium properties but especially for the metastable properties required for the advanced 7-equation two-phase model of RELAP-7. The attainment of accurate properties in an efficient manner with the SBTL method constitutes a completion of a major hurdle in this regard.

Projects are being planned to apply the SBTL method to other fluids, such as heavy water, helium, nitrogen, carbon dioxide etc., and to mixtures, such as humid air.

References

- 1 IAPWS: Guideline on the Fast Calculation of Steam and Water Properties with the Spline-Based Table Look-Up Method (SBTL) (2015), available at <http://www.iapws.org>.
- 2 R.A. Berry, L. Zou, H. Zhao, H. Zhang, J.W. Peterson, R.C. Martineau, S.Y. Kadioglu, D. Andrs, *RELAP-7 Theory Manual*, Idaho National Laboratory technical report INL/EXT-14-31366 (Revision 2) (2016).
- 3 IAPWS: Revised Release on the IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use (2014), available at <http://www.iapws.org>.
- 4 Wagner, W.; and Prueß, A.: The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use, *J. Phys. Chem. Ref. Data* 31, 387-535 (2002).
- 5 IAPWS: Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam (2007), available at <http://www.iapws.org>.

- 6 Wagner, W.; Cooper, J.R.; Dittmann, A.; Kijima, J.; Kretzschmar, H.-J.; Kruse, A.; Mareš, R.; Oguchi, K.; Sato, H.; Stöcker, I.; Šifner, O.; Takaishi, Y.; Tanishita, I.; Trübenbach, J.; and Willkommen, Th.: The IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam, *J. Eng. Gas Turbines & Power* 122, 150-182 (2000).
- 7 IAPWS: Revised Supplementary Release on Backward Equations for Pressure as a Function of Enthalpy and Entropy $p(h,s)$ for Regions 1 and 2 of the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam (2014), available at <http://www.iapws.org>.
- 8 IAPWS: Revised Supplementary Release on Backward Equations for the Functions $T(p,h)$, $v(p,h)$, and $T(p,s)$, $v(p,s)$ for Region 3 of the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam (2014), available at <http://www.iapws.org>.
- 9 IAPWS: Revised Supplementary Release on Backward Equations $p(h,s)$ for Region 3, Equations as a Function of h and s for the Region Boundaries, and an Equation $T_{\text{sat}}(h,s)$ for Region 4 of the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam (2014), available at <http://www.iapws.org>.
- 10 IAPWS: Revised Supplementary Release on Backward Equations for Specific Volume as a Function of Pressure and Temperature $v(p,T)$ for Region 3 of the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam (2014), available at <http://www.iapws.org>.
- 11 Kunick, M.; Kretzschmar, H.-J.; Gampe, U.; di Mare, F.; Hrubý, J.; Duška, M.; Vinš, V.; Singh, A.; Miyagawa, K.; Weber, I.; Pawellek, R.; Novy, A.; Blangetti, F.; Wagner, W.; Lemmon, E.W.; Friend, D.G.; and Harvey, A.H.: Fast Calculation of Steam and Water Properties with the Spline-Based Table Look-Up Method (SBTL), *J. Eng. Gas Turbines & Power*, in preparation.
- 12 Kunick, M.: Fast Calculation of Thermophysical Properties in Extensive Process Simulations with the Spline-Based Table Look-Up Method (SBTL), *VDI Fortschritt-Berichte*, in preparation.
- 13 Kunick, M.; Kretzschmar, H.-J.; di Mare, F.; and Gampe, U.: CFD Analysis of Steam Turbines with the IAPWS Standard on the Spline-Based Table Look-Up Method (SBTL) for the Fast Calculation of Real Fluid Properties, *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition, Vol. 8: Microturbines, Turbochargers and Small Turbomachines; Steam Turbines*, ISBN: 978-0-7918-5679-6 (2015).
- 14 IAPWS: Release on the IAPWS Formulation 2008 for the Viscosity of Ordinary Water Substance (2008), available at <http://www.iapws.org>.
- 15 IAPWS: Release on the IAPWS Formulation 2011 for the Thermal Conductivity of Ordinary Water Substance (2011), available at <http://www.iapws.org>.
- 16 Wagner, W.; and Overhoff, U.: *Extended IAPWS-IF97 Steam Tables*, Springer-Verlag (2006).
- 17 Lemmon, E.W.; Huber, M.L.; and McLinden, M.O.: NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties - REFPROP, National Institute of Standards and Technology, Standard Reference Data Program, Gaithersburg (2013).
- 18 Kunick, M.; and Kretzschmar, H.-J.: *FluidSplines – Software for Generating Spline Functions* (2017), information available at <http://www.thermodynamics-zittau.de>.
- 19 D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie, “MOOSE: A parallel computational framework for coupled systems of nonlinear equations,” *Nuclear Engineering and Design* 239, 1768-1778 (2009).

- 20 R.A. Berry, *Some Specific CASL Requirements of Advanced Multiphase Flow Simulation of Light Water Reactors*, Idaho National Laboratory technical report INL/EXT-10-20529 (2010).
- 21 R.A. Berry, R. Saurel, O. Le Métayer, "The discrete equation method (DEM) for fully compressible, two-phase flows in ducts of spatially varying cross-section," *Nuclear Engineering and Design* 240, 3797-3810 (2010).
- 22 R.A. Berry, "A logical progression of steps for implementation and testing of the 7-equation, two-phase model into a computational framework," *Int. Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C2013)*, Sun Valley, ID, (American Nuclear Society, LaGrange Park, IL USA) (2013).
- 23 R.A. Berry, *Notes on Well-Posed Ensemble Averaged Conservation Equations for Multiphase, Multi-Component, and Multi-Material Flows*, Idaho National Laboratory technical report INL/EXT-05-00516-modified (2005).
- 24 R.A. Berry, R. Saurel, F. Petitpas, E. Daniel, O. Le Métayer, S. Gavrilyuk, N. Dovetta, R.C. Martineau, *Progress in the Development of Compressible, Multiphase Flow Modeling Capability for Nuclear Reactor Flow Applications*, Idaho National Laboratory technical report INL/EXT-08-15002 (2008).
- 25 Richard Saurel, Fabien Petitpas, Ray A. Berry, "Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures," *J. of Computational Physics* 228, 1678-1712 (2009).
- 26 O. Le Métayer, J. Massoni, R. Saurel, "Elaborating equations of state of a liquid and its vapor for two-phase flow models," *Int. J. of Thermal Sciences* 43, 265-276 (2004).

The authors of this contribution

Dipl.-Ing. (FH) Matthias Kunick
m.kunick@hszg.de
Zittau/Görlitz University of Applied Sciences
Faculty of Mechanical Engineering
Department of Technical Thermodynamics
Theodor-Körner-Allee 16
02763 Zittau, Germany
+49-3583-612-4858

Dr. Ray A. Berry, Distinguished Staff Scientist/Engineer
ray.berry@inl.gov
Idaho National Laboratory
Nuclear Science & Technology
P.O. Box 1625
Idaho Falls, Idaho USA 83415-3840
+1-208-526-1254

Prof. Dr.-Ing. habil. Hans-Joachim Kretzschmar
hj.kretzschmar@hszg.de
Zittau/Görlitz University of Applied Sciences
Faculty of Mechanical Engineering
Department of Technical Thermodynamics
Theodor-Körner-Allee 16
02763 Zittau, Germany
+49-3583-612-4846

Prof. Dr.-Ing. Uwe Gampe
uwe.gampe@tu-dresden.de
Dresden University of Technology
Faculty of Mechanical Science and Engineering
Chair of Thermal Power Machinery and Plants
Helmholtzstraße 14
01069 Dresden, Germany
+49-351-463-34491